

EXHIBIT E

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellant: Frankland, Richard et al.)	Examiner: Cheung, Mary Da Zhi
)	Wang
Serial No.: 09/797,488)	
)	Art Unit: 3694
Filed: March 1, 2001)	
)	Our Ref: B-5746CONT 952776-6
For: "INTEGRATED CHANGE MANAGEMENT SYSTEM")	
)	Date: August 28, 2007
)	
)	Re: <i>Appeal to the Board of Appeals</i>

BRIEF ON APPEAL

Mail Stop Appeal-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is an appeal from the Final rejection, dated February 28, 2007, for the above identified patent application. A credit card deposit in the amount of \$500.00 for the fee set forth in 37 C.F.R. 1.17(c) for submitting this Brief and a credit card deposit in the amount of \$120.00 for the time-extension fee set forth in 37 C.F.R. 1.17(a)(1) will be electronically submitted. The Appellants submit that this Appeal Brief is being timely filed within the one-month extended time limit, since the Notice Of Appeal was received on May 29, 2007.

REAL PARTY IN INTEREST

The present application has been assigned to Alternative Systems, Inc. of Half Moon Bay, CA.

Application Serial No. 09/797,488

Appeal brief

Page 2

RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences related to the present application.

STATUS OF CLAIMS

The present Application comprises claims 2, 4-9, 11-24, 26-31, 33-46, 48-52 and 54-66, which all stand rejected.

Claims 1, 3, 10, 25, 32, 47, 53 and 67-155 were cancelled without prejudice.

Claims 2, 4-9, 11-24, 26-31, 33-46, 48-52 and 54-66 are the subject of this appeal and are reproduced in the accompanying claim appendix.

STATUS OF AMENDMENTS

There are no amendments pending in the present application.

SUMMARY OF CLAIMED SUBJECT MATTER

The invention described and claimed in the present application relates generally to the integrated management of information affected by regulatory changes as well as non-regulatory changes. Regulatory changes include changes in environmental, health and safety laws (page 1, lines 4-6; paragraph [0001]).

A number of rules such as federal, state and local laws, statutes, ordinances and regulations control industrial and commercial activities (page 1, lines 8-10; paragraph [0002]). Some rules have overlapping jurisdiction and are not always consistent with one another. Further, the rules and related constraints are constantly changing (page 3, lines 2-9; paragraph [0004]). Various attempts have been made to manage regulatory compliance, but no satisfactory solution has been developed (page 12, lines 24-30; paragraph [0025]). There is accordingly a need for a system that would, for a selected area of commercial or industrial

Application Serial No. 09/797,488
Appeal brief
Page 3

activity: store or access all relevant available information and changes used in connection with the activity; generate and archive records of software system versions used for: data entry, reporting, processing, analysis and results presentation, and changes to these versions; generate all documents and reports required for compliance under applicable regulations, laws and statutes; and provide appropriate user interface, without requiring (re)programming of underlying software (page 13, lines 16-30; paragraph [0027]).

The invention meets the above needs with an integrated system that (1) provides one or more databases that contain information on operations and requirements concerning an activity or area of business; (2) monitors and evaluates the relevance of information on regulatory and non-regulatory changes that affect operations of the business and/or information management requirements; (3) converts the relevant changes into changes in work/task lists, data entry forms, reports, data processing, analysis and presentation (by printing, electronic display, network distribution and/or physical distribution) of data processing and analysis results to selected recipients without requiring the services of one or more programmers to re-program and/or recode the software items affected by the change; and (4) implements receipt of change information and dissemination of data processing and analysis results using the facilities of a network, such as the Internet (page 14, lines 2-14; paragraph [0028]).

In particular, independent claim 2 recites: *"A system for providing a dynamically generated application having one or more functions and one or more user interface elements; comprising:*

a server computer (for example as in a multi-tier server based model, see page 20, lines 14-15; paragraph [0050]);

Application Serial No. 09/797,488
Appeal brief
Page 4

one or more client computers connected to the server computer over a computer network (for example as in a multi-tier server based model, see page 20, lines 14-15; paragraph [0050]);

a first layer (for example a business content layer, see page 21, lines 1-11; paragraph [0054]) associated with the server computer containing information about the unique aspects of a particular application;

a second layer (for example a metadata layer, see page 21, lines 13-20) associated with the server computer containing information about the user interface and functions common to a variety of applications, a particular application being generated based on the data in both the first and second layers;

a third layer (for example a Java data management layer, see page 24, lines 23-26; paragraph [0069]) associated with the server computer that retrieves the data in the first and second layers in order to generate the functionality and user interface elements of the application; and

a change management layer (page 28, lines 4-16; paragraph [0085]) for automatically detecting (box 21 in Fig. 2) changes that affect an application,

each client computer further comprising a browser application (page 26, lines 3-6; paragraph [0075]) being executed by each client computer, wherein a user interface and functionality for the particular application is distributed to the browser application and dynamically generated when the client computer connects to the server computer (page 26, lines 19-22; paragraph [0078]).

Also, independent claim 24 recites:

“A method for dynamically generating an application using a server computer and one or more client computers connected to the server computer over a computer network (for example as in a multi-tier server based model, see page 20, lines 14-15; paragraph

Application Serial No. 09/797,488
Appeal brief
Page 5

[0050]), *the method comprising:*

providing a first layer (for example a business content layer, see page 21, lines 1-11; paragraph [0054]) containing information about the unique aspects of a particular application;

providing a second layer (for example a metadata layer, see page 21, lines 13-20) containing information about the user interface and functions common to a variety of applications, wherein a particular application is generated based on the data in the first and second layers;

establishing a connection between a client computer and the server computer (see page 20, lines 14-15; paragraph [0050]);

providing a third layer (for example a Java data management layer, see page 24, lines 23-26; paragraph [0069]) that retrieves the data in the first and second layers in order to generate the functionality and user interface for a particular application for the client computer as the client computer connects to the server computer;

automatically detecting (box 21 in Fig. 2) changes that affect a particular application (page 28, lines 4-6; paragraph [0085]); and

distributing the user interface and functionality of the particular application to the client computer wherein the particular application and its user interface are dynamically re-generated each time a client establishes a connection with the server computer (page 26, lines 19-22; paragraph [0078])”.

Further, independent claim 46 recites:

“A server for dynamically generating an application for one or more client computers connected to the server computer by a computer network (for example as in a multi-tier server based model, see page 20, lines 14-15; paragraph [0050]), comprising:

Application Serial No. 09/797,488

Appeal brief

Page 6

a first layer (for example a business content layer, see page 21, lines 1-11; paragraph [0054]) associated with the server containing information about the unique aspects of a particular application;

a second layer (for example a metadata layer, see page 21, lines 13-20) associated with the server containing information about the user interface and functions common to a variety of applications;

a third layer (for example a Java data management layer, see page 24, lines 23-26; paragraph [0069]) that retrieves the data in the first and second layers in order to generate functionality and user interface elements of the application;

a change management layer (page 28, lines 4-16; paragraph [0085]) for automatically detecting (box 21 in Fig. 2) changes that affect an application;

means for dynamically generating (page 26, lines 19-22; paragraph [0078]) a particular application based on the first and second layers each time a client computer connects to the server computer; and

means for distributing the user interface and functionality of the particular application to a client computer (page 26, lines 3-6; paragraph [0075])”.

* * *

Application Serial No. 09/797,488
Appeal brief
Page 7

GROUND OF REJECTION TO BE REVIEWED ON APPEAL

Issue 1: Whether claims 2, 4-9, 11-24, 26-31, 33-46, 48-52 and 54-66 are patentable under 35 U.S.C. 102(e) over U.S. Pat. No. 5,960,200 to Eager.

ARGUMENT

Issue I: Whether claims 2, 4-9, 11-24, 26-31, 33-46, 48-52 and 54-66 are patentable under 35 U.S.C. 102(e) over U.S. Pat. No. 5,960,200 to Eager (hereinafter “Eager”).

Rejection of claim 2

In a non-final action issued on September 5, 2006, the Examiner rejected claim 2 as being anticipated by U.S. 5,960,200 to Eager, under the rationale that:

-a business process layer taught by Eager reads on the first layer recited in claim 2;

-a functionality layer taught by Eager reads on the second layer recited in claim 2;

-a presentation layer taught by Eager reads on the third layer recited in claim 2; and

-a re-architecting system taught by Eager reads on the change management layer recited in claim 2.

Application Serial No. 09/797,488
Appeal brief
Page 8

In response to this previous action, the Appellants disagreed with the rationale of the Examiner, and argued that the business process layer (120) and functionality layer (130) of Eager are distinct from a data storage layer (150), so that they cannot read on the recited first and second layers “containing information”.

In the Final action issued on February 28, 2006, the Examiner opines that the arguments above are not persuasive.

In particular, in response to the Appellant’s arguments that Eager fails to teach the first layer containing information about the unique aspects of particular application, the Examiner opines that “Eager teaches a business process layer that corresponds to the first layer as claimed by the applicant (column 10, lines 33-41 and Fig. 1), and the business layer provides initial application menu or screen and initial objects to the interface engine (column 10, lines 38-41), in which the initial application menu or screen and initial objects correspond to the information contained in the first layer”. The Examiner also opines that “although Eager does not explicitly disclose how long the information is contained in the business layer, the information could either be stored in the business layer in short period of time or in a long period of time; and in either case, Eager’s teaching reads on the claimed language”.

Also, in response to the Appellant’s arguments that Eager fails to teach a second layer that contains information about the user interface and functions common to a variety of applications, the Examiner opines that “Eager teaches a functionality layer that corresponds to the second layer (column 12, lines 12-42 and Figs. 1, 8) and the information contained by the second layer is taught by Eager as the result information is generated by the functionality layer and later is

Application Serial No. 09/797,488

Appeal brief

Page 9

transferred to the business layer or the state router (column 12, lines 18-21)". The Examiner further opines that "although Eager does not explicitly disclose how long the result information is contained in the functionality layer, the result information could either be stored in the functionality layer in short period of time or in a long period of time; and in either case, Eager's teachings reads on the claimed language".

The Appellants respectfully submit that the interpretation the Examiner makes of the term "containing" is broader than the 'broadest reasonable interpretation' that one skilled in the art would make in light of the specification. Indeed, according to this interpretation, any single means through which data transit would read on means "containing" the data. Thus, for example a data bus would be considered as means containing data. However, one skilled in the art knows well that a data bus is a means that carries data, but does not contain data.

At least in view of the above, the Appellants submit that the interpretation the Examiner makes of Eager does not comply with section 2111.01(II) of the M.P.E.P, and request that the Examiner's rejection of claim 2 be overturned.

The Appellants further note that, even assuming, *arguendo*, that the interpretation the Examiner makes of the term "containing" were not broader than its broadest reasonable interpretation, Eager would still be very different from the invention as recited in claim 2.

Eager relates (see claim 1 of Eager) to an automated system for converting, or transitioning, legacy applications operable on a legacy computing system to a distributed infrastructure on a multi-tiered computer architecture. Eager provides, in particular, for "*an automated converter to transition the legacy application to a target application operable on the multi-tiered computer architecture*".

Application Serial No. 09/797,488
Appeal brief
Page 10

However, Eager teaches automatically converting an application operable in a legacy system to an application operable in a multi-tiered system. Nowhere does Eager disclose or suggest any feature for monitoring applications and detecting changes affecting the application, and in particular a change management layer for "*automatically detecting changes that affect an application*" as recited in claim 2.

In section 4 of the Final action of February 28, 2007, the Examiner opines that Eager shows "(f) a change management layer for automatically detecting changes that affect an application (column 2, lines 34-57 and column 4, lines 42-51 and Figs. 1, 16, 17, 24; specifically, 'change management layer' corresponding to re-architecting system in Eager's teaching)". The Appellants respectfully disagree.

As to what is taught in column 2, lines 34-57 of Eager: the Appellants note that column 2, lines 34-57 of Eager recite that "*For each source application, there is a range of available transition choices, including the option of translating the source application to the new target architecture without changing any of the existing functionality and the option of re-engineering the source application by changing the existing functionality. The source application may also be replaceable by a commercial product or a custom application written in-house. The source applications are then transitioned in order of priority to the new architecture*".

This passage teaches that Eager allows translating source applications to the new target architecture without changing the existing functionality. It also teaches that Eager allows re-engineering the source application by changing the existing functionality.

However, this passage does not relate to detecting changes in the

Application Serial No. 09/797,488

Appeal brief

Page 11

applications, and relates even less to automatically detecting such changes.

Column 2, lines 34-57 of Eager further recites, *“Specifically, a preferred system in accordance with the present invention includes the automated capability to translate existing source applications into new target applications on a multi-tiered client/server architecture. The translation of source applications to target applications includes the conversion of user interfaces, procedural languages, and data definitions. These conversions use a two-phase process where source program components written in the source languages are first translated to components in a common intermediate language. The intermediate language components are then translated to target program components in the target languages. By using a common intermediate language, only one translation module is required for each source and target language”*.

This passage teaches that Eager allows automatically translating sources application.

This passage does not relate to detecting changes in the applications, or to automatically detecting such changes.

As to what is taught in Column 4, lines 42-51 of Eager: column 4, lines 42-51 recites that *“a preferred re-architecting system 20 includes a user interface conversion utility 210, a procedural language conversion utility 220, and a data definition language conversion utility 230. The procedural language conversion utility 220 is in communication with the functionality layer 130 and the data access layer 140 of the multi-tier architecture 10. The user interface conversion utility 210 is in communication with the user interface repository 152 and the data definition language conversion utility 230 is in communication with the data record repository 158”*.

This passage discloses a preferred structure of a re-architecting system of Eager, but does not disclose or suggest any feature that would allow

Application Serial No. 09/797,488
Appeal brief
Page 12

“automatically detecting changes that affect an application” as recited in claim 2.

Further, the Appellants note that Eager discloses in column 6, lines 50-52, that *“the re-architect option translates the existing application to the new architecture without changing any of the existing functionality”*, and thus explicitly and unambiguously discloses that the “re-architecting” of Eager is not related to implementing (or detecting) any change that affects an application.

As to what is taught in Figure 1 of Eager: Figure 1 of Eager shows a re-architecting system 20 comprising *“a user interface conversion utility 210, a procedural language conversion utility 220, and a data definition language conversion utility 230”*.

The Appellants note that the Examiner has failed to show which of the features of system 20 would read on any feature that would allow *“automatically detecting changes that affect an application”* as recited in claim 2. Further, as detailed above, Eager discloses that the system’s “re-architecting” is not related to handling changes in an application.

As to what is taught in Figure 16 of Eager: Figure 16 of Eager is a block diagram of the user interface conversion utility 210 of Fig. 1, which *“converts the user interface of an existing application represented by the source user interface definitions 211 into target user interface definitions 213 using the user interface converter 212”* (col. 23, lines 4-8).

As detailed above, Fig. 1 of Eager relates to a re-architecting system, wherein such re-architecting is done *“without changing any of the existing functionality”*. The Appellants note that the Examiner has failed to explain how a

Application Serial No. 09/797,488
Appeal brief
Page 13

portion of a system provided for implementing an operation devoid of change would disclose or suggest means for detecting changes, and in particular means for “*automatically detecting changes that affect an application*” as recited in claim 2.

As to what is taught in Figure 17 of Eager: Figure 17 of Eager is a flow diagram wherein “*the procedural language conversion utility 220 converts the functionality and data access programs of an existing application into the programming language targeted for the implementation of the functionality layer 130 (Fig. 1). This conversion process consists of two main phases. A first phase (Phase A) converts the source language 221 into an intermediary language 225. A second phase (Phase B) then transforms the intermediary language 225 into the final target language 227*” (col. 24, lines 45-53).

The Appellants note that Figure 17 only relates to converting a source application in a source language into a target application in a target language, and note that the Examiner has failed to show which part of the feature of figure 17 would read on means for “*automatically detecting changes that affect an application*” as recited in claim 2.

As to what is taught in Figure 24 of Eager: Figure 24 of Eager is a block diagram of the data definition language conversion utility 230 of Fig. 1 that is “*used to convert a source database language 231 into a target database language 237 using a database converter 234*” (col. 28, lines 11-14).

The Appellants note that Figure 24 only relates to converting a source language into a target language, and note that the Examiner has failed to show which part of the feature of figure 24 would read on means for “*automatically detecting changes that affect an application*” as recited in claim 2.

Application Serial No. 09/797,488
Appeal brief
Page 14

In addition to the above remarks, the Appellants note that the only mention of a change in an application is made in relation to GUI files 248, that are “*used by application developers and maintenance personnel to modify application screens and messages as part of the re-engineering system 30*” (col. 23, line 66 – col. 24, line 1).

The Appellants note that Eager teaches modifying applications in relation to Eager’s re-engineering system, and not Eager’s re-architecting system as opined by the Examiner.

Further, the Appellants note that Eager explicitly teaches that in the re-engineering system, it is “*application developers and maintenance personnel*” that “*modify application screens and messages*”, thus teaching away from any means that would operate “*automatically*”, and in particular that would allow “*automatically detecting changes that affect an application*” as recited in claim 2.

In conclusion, the Appellants note that Eager relates to means for automatically converting a source system into a target system, wherein:

- a re-architecting option allows translating applications in the source system “*without changing any of the existing functionality*”; and

- a re-engineering option allows “*application developers and maintenance personnel*” to “*modify application screens and messages*”, as seen above in relation to Figure 16.

Contrary to the Examiner’s assertion, no part of Eager seems to disclose or suggest any such thing as means “*for automatically detecting changes that affect an application*” or “*a change management layer for automatically detecting changes that affect an application*” as recited in claim 2.

Application Serial No. 09/797,488
Appeal brief
Page 15

The Appellants note that when converting a source application into a target application, the converter of Eager parses (with parser 243) the language of the source application to convert it in a target application. If the source application has changed, one can assume that such change will be introduced in the target application. However, the Appellants note that there is no suggestion in Eager that the parsing of a source application having non-changed instruction as well as changed instructions would distinguish in any way the changed instructions from the non-changed instructions. Accordingly, parsing the source applications cannot be deemed to disclose or suggest “detecting” changes that affect an application. For this reason also, Eager cannot be argued to disclose or suggest “*a change management layer for automatically detecting changes that affect an application*” as recited in claim 2.

The Appellants respectfully submit that at least in view of the above, claim 2 is patentable over Eager, and respectfully request that the Examiner’s rejection of claim 2 be properly overturned.

Rejection of claims 24 and 46

The Appellants respectfully submit that the arguments used above for showing that claim 2 is patentable over Eager can also be used to show that Eager does not disclose or suggest a method as recited in claim 24, and in particular comprising: “*automatically detecting changes that affect a particular application*” or a server as recited in claim 46, and in particular comprising “*a change management layer for automatically detecting changes that affect an application*”.

Application Serial No. 09/797,488
Appeal brief
Page 16

The Appellants respectfully submit that at least in view of the above, claims 24 and 46 are patentable over Eager, and respectfully request that the Examiner's rejection of claims 24 and 46 be properly overturned.

Rejection of claims 4-9, 11-23, 26-31, 33-45, 48-52 and 54-66

Claims 4-9 and 11-23 depend directly or indirectly on claim 2; claims 26-31 and 33-45 depend directly or indirectly on claim 2; and claims 48-52 and 54-66 depend directly or indirectly on claim 46.

Appellants note that "if an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious." *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988).

The Appellants respectfully submit that at least in light of the above discussion, dependent claims 4-9, 11-23, 26-31, 33-45, 48-52 and 54-66 are also allowable.

Further, as developed hereafter, the Appellants disagree with the specific rationale developed by the Examiner for rejecting the dependent claims.

Rejection of claim 4

In section 4 of the Final action of February 28, 2007, the Examiner opines that "as to claim 4, Eager teaches the third layer further comprises a Java data management layer having means for distributing one or more JAVA applets to the client computer wherein the JAVA applets dynamically generate and present the user interface and functionality to the user based on the first and second

Application Serial No. 09/797,488

Appeal brief

Page 17

layers (column 8, lines 3-22 and column 10, lines 7-26)". The Appellants respectfully disagree with the Examiner.

Column 8, lines 3-22 of Eager recites that *"an Open Systems Integration (OSI) process 469 is a manual implementation process that focuses on integrating applications that are purchased, newly custom developed, re-architected, or re-engineered to share data and screens. This process includes the definition of business goals and objectives, the definition of applicable business processes, the study of application interactions and data relationships, and the planning of hardware and software infrastructures. The OSI process 469 also includes the implementation of the integration, including detailed implementation plan and schedule and detailed requirements and design documentation, user acceptance testing, comprehensive technical documentation, and empowerment of support staff for the maintenance phase. One powerful example of integration at the user interface layer using the OSI process 469 is the creation of a corporate intranet using internet Hyper-Text Manipulation Language (HTML) or a highly-level language generating HTML, such as Java from Sun Microsystems to provide a user-friendly, platform independent, common user interface to corporate application"*.

The Appellants note that the above excerpt does not recite the words "JAVA applet", and that the Examiner has failed to explain where the above excerpt would suggest using JAVA applets. It follows that the Examiner has also failed to explain where the above excerpt would suggest JAVA applets that *"dynamically generate and present" "user interface and functionality"* to a user" based on first and second layers.

Column 10, lines 7-26 of Eager recites that *"FIG. 5 is a schematic diagram of a sample mapping between application user interface representation structures 116 and display platform user interface representation structures 118. In the figure, the user interface display platform 115 is exemplified as Microsoft Windows 3.x and the display*

Application Serial No. 09/797,488

Appeal brief

Page 18

platform user interface representation structures 117 are thus the internal Windows 3.x management structures. However, other user interface display platforms 115 using similar internal structures to manage windows are supported by the exact same user interface engine 117. Notably, the Internet's world-wide web, based on the HTML or Java user interface languages, is another example of user interface display platform 115. Indeed, in a preferred embodiment of the present invention, the user interface engine 117 is written using Microsoft Visual C++ and based on the industry-standard Microsoft Foundations Classes (MFC) class library, which allows cross-platform development for Windows 3.x, Windows 95, Windows NT, MacOS, and UNIX-based user interface display platforms 115, including internet web servers".

The Appellants note that the above excerpt does not recite the words "JAVA applet", and that the Examiner has failed to explain where the above excerpt would suggest using JAVA applets. It follows that the Examiner has also failed to explain where the above excerpt would suggest JAVA applets that "dynamically generate and present" "user interface and functionality" to a user" based on first and second layers.

At least in view of the above, the Appellants respectfully submit that the Examiner has failed to show that Eager discloses or suggest the features recited in claim 4, and in particular "data management layer having means for distributing one or more JAVA applets to the client computer wherein the JAVA applets dynamically generate and present the user interface and functionality to the user based on the first and second layers".

The Appellants respectfully submit that, for this reason also, claim 4 is patentable over Eager.

Application Serial No. 09/797,488
Appeal brief
Page 19

Rejection of claims 26 and 48

Claim 26 depends on claim 24 and recites features corresponding to the features of claim 4. Claim 48 depends on claim 46 and recites features corresponding to the features of claim 4.

The Appellants note that the arguments used above to show that claim 4 is patentable over Eager at least because Eager does not disclose or suggest “JAVA applets” as recited in claim 4 can also be used to show that claims 26 and 48 are patentable over Eager.

Rejection of claim 11

In section 4 of the Final action of February 28, 2007, the Examiner opines that “as to claims 11-12, Eager teaches the server computer further comprises the change management layer further comprises one or more intelligent agents that detects changes that affect an application (column 2, lines 34-57 and column 4, lines 42-51 and Figs. 1, 16, 17, 24; specifically, ‘change management layer’ corresponding to re-architecting system in Eager’s teaching)”. The Appellants respectfully disagree.

As detailed above, column 2, lines 34-57 of Eager recites that “*For each source application, there is a range of available transition choices, including the option of translating the source application to the new target architecture without changing any of the existing functionality and the option of re-engineering the source application by changing the existing functionality. The source application may also be replaceable by a commercial product or a custom application written in-house. The source applications are then transitioned in order of priority to the new architecture*”.

Application Serial No. 09/797,488
Appeal brief
Page 20

This passage teaches that Eager allows translating sources application to the new target architecture without changing the existing functionality. It also teaches that Eager allows re-engineering the source application by changing the existing functionality.

However, this passage does not relate to detecting changes in the applications, and relates even less to detecting automatically such changes.

Column 2, lines 34-57 of Eager further recites, *"Specifically, a preferred system in accordance with the present invention includes the automated capability to translate existing source applications into new target applications on a multi-tiered client/server architecture. The translation of source applications to target applications includes the conversion of user interfaces, procedural languages, and data definitions. These conversions use a two-phase process where source program components written in the source languages are first translated to components in a common intermediate language. The intermediate language components are then translated to target program components in the target languages. By using a common intermediate language, only one translation module is required for each source and target language"*.

This passage teaches that Eager allows automatically translating sources application.

Column 2, lines 34-57 of Eager does not disclose or suggest detecting changes in the applications. It follows that contrary to the assertion of the Examiner, column 2, lines 34-57 of Eager does not disclose or suggest means for detecting changes, and in particular such as *"one or more intelligent agents that detect changes that affect an application"* as recited in claim 11.

Similarly, column 4, lines 42-51 of Eager recites that *"a preferred re-architecting system 20 includes a user interface conversion utility 210, a procedural*

Application Serial No. 09/797,488
Appeal brief
Page 21

language conversion utility 220, and a data definition language conversion utility 230. The procedural language conversion utility 220 is in communication with the functionality layer 130 and the data access layer 140 of the multi-tier architecture 10. The user interface conversion utility 210 is in communication with the user interface repository 152 and the data definition language conversion utility 230 is in communication with the data record repository 158".

Further, as detailed above, Eager discloses in column 6, lines 50-52, that "re-architecting" *"translates the existing application to the new architecture without changing any of the existing functionality"*, thus teaching away from the idea of implementing or detecting changes.

It follows that column 4, lines 42-51 of Eager discloses a preferred structure of a re-architecting system of Eager, but does not disclose or suggest detecting changes in the applications. Contrary to the assertion of the Examiner, column 4, lines 42-51 of Eager does therefore not disclose or suggest means for detecting changes, and in particular such as *"one or more intelligent agents that detect changes that affect an application"* as recited in claim 11.

Similarly, Figure 1 of Eager shows a re-architecting system 20 comprising *"a user interface conversion utility 210, a procedural language conversion utility 220, and a data definition language conversion utility 230"*.

As detailed above, the "re-architecting" of Eager is not related to handling changes in an application. The Appellants further note that the Examiner has failed to point out which of the above utilities/features would disclose or suggest means for detecting changes, and in particular such as *"one or more intelligent agents that detect changes that affect an application"* as recited in claim 11.

Application Serial No. 09/797,488
Appeal brief
Page 22

Figure 16 of Eager shows a block diagram of the user interface conversion utility 210 of Fig. 1, which *“converts the user interface of an existing application represented by the source user interface definitions 211 into target user interface definitions 213 using the user interface converter 212”* (col. 23, lines 4-8).

As detailed above, Fig. 1 of Eager relates to a re-architecting system, wherein such re-architecting is done *“without changing any of the existing functionality”*. The Appellants note that the Examiner has failed to explain how a portion of a system provided for implementing an operation devoid of change would disclose or suggest means for detecting changes, and in particular *“intelligent agents”* such as *“one or more intelligent agents that detect changes that affect an application”* as recited in claim 11.

Figure 17 of Eager shows a flow diagram wherein *“the procedural language conversion utility 220 converts the functionality and data access programs of an existing application into the programming language targeted for the implementation of the functionality layer 130 (Fig. 1). This conversion process consists of two main phases. A first phase (Phase A) converts the source language 221 into an intermediary language 225. A second phase (Phase B) then transforms the intermediary language 225 into the final target language 227”* (col. 24, lines 45-53).

The Appellants note that Figure 17 only relates to converting a source application in a source language into a target application in a target language, and note that the Examiner has failed to show which part of the feature of figure 17 would disclose or suggest means for detecting changes, and in particular *“intelligent agents”* such as *“one or more intelligent agents that detect changes that affect an application”* as recited in claim 11.

Application Serial No. 09/797,488
Appeal brief
Page 23

Figure 24 of Eager is a block diagram of the data definition language conversion utility 230 of Fig. 1 that is *“used to convert a source database language 231 into a target database language 237 using a database converter 234”* (col. 28, lines 11-14).

The Appellants note that Figure 24 only relates to converting a source language into a target language, and note that the Examiner has failed to show which part of the feature of figure 24 would disclose or suggest means for detecting changes, and in particular *“intelligent agents”* such as *“one or more intelligent agents that detect changes that affect an application”* as recited in claim 11.

In conclusion, contrary to the Examiner’s assertion, Eager does not disclose or suggest means for detecting changes, and in particular *“intelligent agents”* such as *“one or more intelligent agents that detect changes that affect an application”* as recited in claim 11.

In view of the above also, claim 11 is patentable over Eager.

Rejection of claim 12

In section 4 of the Final action of February 28, 2007, the Examiner opines that *“as to claims 11-12, Eager teaches the server computer further comprises the change management layer further comprises one or more intelligent agents that detects changes that affect an application (column 2, lines 34-57 and column 4, lines 42-51 and Figs. 1, 16, 17, 24; specifically, ‘change management layer’ corresponding to re-architecting system in Eager’s teaching)”*. The Appellants respectfully disagree.

As detailed above in relation to claim 11, none of the above excerpts or

Application Serial No. 09/797,488
Appeal brief
Page 24

figures disclose or even remotely suggest means for detecting changes, and in particular “*intelligent agents*” such as “*one or more intelligent agents that detect changes that affect an application*” as recited in claim 11.

Claim 12 depends on claim 11 and recites “*means for automatically modifying the first and second layers in response to the intelligent agents in order to automatically change the functionality and user interface elements of the application based on the changes detected by the intelligent agents*”.

The Appellants note that since as seen above, Eager and in particular “column 2, lines 34-57 and column 4, lines 42-51 and Figs. 1, 16, 17, 24; specifically, ‘change management layer’ corresponding to re-architecting system in Eager’s teaching” does not disclose or suggest “*intelligent agents that detect changes that affect an application*”.

It follows that Eager and in particular “column 2, lines 34-57 and column 4, lines 42-51 and Figs. 1, 16, 17, 24; specifically, ‘change management layer’ corresponding to re-architecting system in Eager’s teaching” can also not be deemed to disclose or suggest means for operating in response to such intelligent agent, such as the “*means for automatically modifying the first and second layers in response to the intelligent agents in order to automatically change the functionality and user interface elements of the application based on the changes detected by the intelligent agents*” recited in claim 12.

In view of the above also, claim 12 is patentable over Eager.

Rejection of claims 13-21

In section 4 of the Final action of February 28, 2007, the Examiner opines that “as to claims 13-21, Eager teaches a builder module for permitting a user to

Application Serial No. 09/797,488
Appeal brief
Page 25

build a user interface for a particular application using the second layer (see facilitation tools 360 in Figs. 2 and 33). The Appellants respectfully disagree.

Claim 21 recites a builder module that *“comprises an intelligent agent builder for generating the intelligent agents that detect changes associated with the particular business application”*.

The Appellants note that the arguments used above to show that Eager does not disclose or suggest “intelligent agents” as recited in claim 11 can also be used to show that Eager does not disclose or suggest an *“intelligent agent builder”* as recited in claim 21.

In view of the above also, claim 21 is patentable over Eager.

Rejection of claims 33, 34, 43, 54, 55, 64

Claims 33, 34 and 43 depend on claim 24 and recite features corresponding to the features of claims 11, 12 and 21. Claims 54, 55 and 64 depend on claim 46 and recite features corresponding to the features of claims 11, 12 and 21.

The Appellants note that the arguments used above to show that claims 11, 12 and 21 are patentable over Eager at least because Eager does not disclose or suggest “intelligent agents” as recited in claims 11, 12 and 21 can also be used to show that claims 33, 34 and 43 as well as claims 54, 55 and 64 are patentable over Eager.

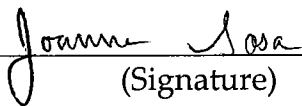
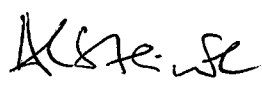
* * *

Application Serial No. 09/797,488
Appeal brief
Page 26

CONCLUSION

For the extensive reasons advanced above, Appellants respectfully contend that each claim is patentable. Therefore, reversal of the above-addressed rejections and objections and re-opening of the prosecution is respectfully solicited.

The Commissioner is authorized to charge any additional fees that may be required or credit overpayment to deposit account no. 12-0415. In particular, if this response is not timely filed, the Commissioner is authorized to treat this response as including a petition to extend the time period pursuant to 37 CFR 1.136(a) requesting an extension of time of the number of months necessary to make this response timely filed and the petition fee due in connection therewith may be charged to deposit account no. 12-0415.

<p>I hereby certify that this correspondence is being electronically filed by EFS-Web in the United States Patent and Trademark Office on</p> <p>August 28, 2007 (Date of Transmission)</p> <p>Joanna Sosa (Name of Person Transmitting)</p> <p> (Signature)</p>	<p>Respectfully submitted,</p> <p></p> <p>Alessandro Steinfel Attorney for the Appellants Reg. No. 28,145 LADAS & PARRY 5670 Wilshire Boulevard, Suite 2100 Los Angeles, California 90036 (323) 934-2300 voice (323) 934-0202 facsimile</p>
---	---

Application Serial No. 09/797,488

Appeal brief

Page 27

8 - 28 - 07	
(Date)	

Attachments: Claims 4-9, 11-23, 26-31, 33-45, 48-52 and 54-66

Application Serial No. 09/797,488
Appeal brief
Page 28

CLAIMS APPENDIX

1. (canceled)
2. A system for providing a dynamically generated application having one or more functions and one or more user interface elements; comprising:
 - a server computer;
 - one or more client computers connected to the server computer over a computer network;
 - a first layer associated with the server computer containing information about the unique aspects of a particular application ;
 - a second layer associated with the server computer containing information about the user interface and functions common to a variety of applications, a particular application being generated based on the data in both the first and second layers;
 - a third layer associated with the server computer that retrieves the data in the first and second layers in order to generate the functionality and user interface elements of the application; and
 - a change management layer for automatically detecting changes that affect an application,
 - each client computer further comprising a browser application being executed by each client computer, wherein a user interface and functionality for the particular application is distributed to the browser application and

Application Serial No. 09/797,488

Appeal brief

Page 29

dynamically generated when the client computer connects to the server computer.

3. (canceled)

4. The system of Claim 2, wherein the third layer further comprises a JAVA data management layer having means for distributing one or more JAVA applets to the client computer wherein the JAVA applets dynamically generate and present the user interface and functionality to the user based on the first and second layers.

5. The system of Claim 2, wherein the second layer comprises a business content database having data about one or more different predetermined business applications.

6. The system of Claim 5, wherein the data further comprises one or more of business knowledge, logical designs, physical designs, physical structures and relationships associated with the predetermined business application.

7. The system of Claim 5, wherein the second layer comprises a metadata database comprising data about the structures and functions associated with any application.

8. The system of Claim 7, wherein the metadata database further comprises data about the user interface and functionality including one or more of tools, worklists, data entry forms, reports, documents, processes, formulas and images.

Application Serial No. 09/797,488

Appeal brief

Page 30

9. The system of Claim 2, wherein each client computer further comprises a JAVA enabled web browser to permit remote user access.

10. (canceled)

11. The system of Claim 2, wherein the change management layer further comprises one or more intelligent agents that detect changes that affect an application.

12. The system of Claim 11, wherein the server further comprises means for automatically modifying the first and second layers in response to the intelligent agents in order to automatically change the functionality and user interface elements of the application based on the changes detected by the intelligent agents.

13. The system of Claim 2, wherein the server further comprises a builder module for permitting a user to build a user interface for a particular application using the second layer.

14. The system of Claim 13, wherein the builder module further comprises a form builder for one or more of editing an existing form and generating a new form that contains the data for a particular application.

15. The system of Claim 13, wherein the builder module further comprises an event builder for generating triggering events for a form.

Application Serial No. 09/797,488
Appeal brief
Page 31

16. The system of Claim 13, wherein the builder module further comprises a report builder for building a report for a particular application.

17. The system of Claim 13, wherein the builder module further comprises a document builder for mapping a document onto the first layer.

18. The system of Claim 13, wherein the builder module further comprises a formula builder for generating formulas.

19. The system of Claim 16, wherein the builder module further comprises a view/query builder for generating one or more views/queries used in the reports.

20. The system of Claim 13, wherein the builder module further comprises a worklist builder for generating a worklist.

21. The system of Claim 13, wherein the builder module further comprises an intelligent agent builder for generating the intelligent agents that detect changes associated with the particular business application.

22. The system of Claim 2, wherein the first and second layers are stored on the server computer.

23. The system of Claim 2, wherein the first and second layers are distributed across one or more server computers.

Application Serial No. 09/797,488

Appeal brief

Page 32

24. A method for dynamically generating an application using a server computer and one or more client computers connected to the server computer over a computer network, the method comprising:

providing a first layer containing information about the unique aspects of a particular application;

providing a second layer containing information about the user interface and functions common to a variety of applications, wherein a particular application is generated based on the data in the first and second layers;

establishing a connection between a client computer and the server computer;

providing a third layer that retrieves the data in the first and second layers in order to generate the functionality and user interface for a particular application for the client computer as the client computer connects to the server computer;

automatically detecting changes that affect a particular application; and

distributing the user interface and functionality of the particular application to the client computer wherein the particular application and its user interface are dynamically re-generated each time a client establishes a connection with the server computer.

25. (canceled)

26. The method of Claim 24, wherein the third layer further comprises a JAVA data management layer that distributes one or more JAVA applets to the client computer wherein the JAVA applets dynamically generate and present the user interface and functionality to the user based on the first and second layers.

Application Serial No. 09/797,488

Appeal brief

Page 33

27. The method of Claim 24, wherein the first layer comprises a business content database having data about one or more different predetermined business applications.

28. The method of Claim 27, wherein the data further comprises one or more of business knowledge, logical designs, physical designs, physical structures and relationships associated with the predetermined business application.

29. The method of Claim 27, wherein the second layer comprises a metadata database comprising data about the structures and functions associated with any application.

30. The method of Claim 29, wherein the metadata database further comprises data about the user interface including one or more of tools, worklists, data entry forms, reports, documents, processes, formulas and images.

31. The method of Claim 24, wherein each client computer further comprises a JAVA enabled web browser to permit remote user access.

32. (canceled)

33. The method of Claim 24, wherein the change management layer further comprises one or more intelligent agents that detect changes that affect an application.

Application Serial No. 09/797,488

Appeal brief

Page 34

34. The method of Claim 33, wherein automatically detecting changes further comprises automatically modifying the first and second layers in response to the intelligent agents in order to automatically change the functionality and user interface elements of the application based on the changes detected by the intelligent agents.

35. The method of Claim 24 further comprising permitting a user to build a user interface for a particular application using the second layer.

36. The method of Claim 35, wherein the building further comprises one or more of editing an existing form and generating a new form that contains the data for a particular application.

37. The method of Claim 35, wherein the building further comprises generating triggering events for a form.

38. The method of Claim 35, wherein the building further comprises building a report for a particular application.

39. The method of Claim 35, wherein the building further comprises mapping a document onto the first layer.

40. The method of Claim 35, wherein the building further comprises generating formulas associated with the application.

Application Serial No. 09/797,488
Appeal brief
Page 35

41. The method of Claim 38, wherein the building further comprises generating one or more views/queries used in the reports.

42. The method of Claim 35, wherein the building further comprises generating a worklist.

43. The method of Claim 35, wherein the building further comprises generating the intelligent agents that detect changes associated with the particular business application.

44. The method of Claim 24 further comprising distributing the first and the second layers on the server computer.

45. The method of Claim 24 further comprising distributing the first and second layers across one or more server computers.

46. A server for dynamically generating an application for one or more client computers connected to the server computer by a computer network, comprising:

- a first layer associated with the server containing information about the unique aspects of a particular application;

- a second layer associated with the server containing information about the user interface and functions common to a variety of applications;

- a third layer that retrieves the data in the first and second layers in order to generate functionality and user interface elements of the application;

Application Serial No. 09/797,488

Appeal brief

Page 36

a change management layer for automatically detecting changes that affect an application;

means for dynamically generating a particular application based on the first and second layers each time a client computer connects to the server computer; and

means for distributing the user interface and functionality of the particular application to a client computer.

47. (canceled)

48. The server of Claim 46, wherein the third layer further comprises a JAVA data management layer having means for distributing one or more JAVA applets to the client computer wherein the JAVA applets dynamically generate and present the user interface and functionality to the user based on the first and second layers.

49. The server of Claim 46, wherein the first layer comprises a business content database having data about one or more different predetermined business applications.

50. The server of Claim 49, wherein the data further comprises one or more of business knowledge, logical designs, physical designs, physical structures and relationships associated with the predetermined business application.

51. The server of Claim 49, wherein the second layer comprises a metadata

Application Serial No. 09/797,488

Appeal brief

Page 37

database comprising data about the structures and functions associated with any application.

52. The server of Claim 51, wherein the metadata database further comprises data about the user interface including one or more of tools, worklists, data entry forms, reports, documents, processes, formulas and images.

53. (canceled)

54. The server of Claim 46, wherein the change management layer further comprises one or more intelligent agents that detect changes that affect an application.

55. The server of Claim 54, wherein the change management layer further comprises means for automatically modifying the first and second layers in response to the intelligent agents in order to automatically change the functionality and user interface elements of the application based on the changes detected by the intelligent agents.

56. The server of Claim 46, wherein the server further comprises a builder module for permitting a user to build a user interface for a particular application using the second layer.

57. The server of Claim 56, wherein the builder module further comprises a form builder for one or more of editing an existing form and generating a new form that contains the data for a particular application.

Application Serial No. 09/797,488

Appeal brief

Page 38

58. The server of Claim 56, wherein the builder module further comprises an event builder for generating triggering events for a form.

59. The server of Claim 56, wherein the builder module further comprises a report builder for building a report for a particular application.

60. The server of Claim 56, wherein the builder module further comprises a document builder for mapping a document onto the first layer.

61. The server of Claim 56, wherein the builder module further comprises a formula builder for generating formulas.

62. The server of Claim 59, wherein the builder module further comprises a view/query builder for generating one or more views/queries used in the reports.

63. The server of Claim 56, wherein the builder module further comprises a worklist builder for generating a worklist.

64. The server of Claim 56, wherein the builder module further comprises an intelligent agent builder for generating the intelligent agents that detect changes associated with the particular business application.

65. The server of Claim 46, wherein the first and second layers are distributed on the server computer.

Application Serial No. 09/797,488

Appeal brief

Page 39

66. The server of Claim 46, wherein the first and second layers are distributed across one or more server computers.

67. – 155. (canceled)

Application Serial No. 09/797,488

Appeal brief

Page 40

EVIDENCE APPENDIX

There is no evidence submitted with the present Appeal Brief.

Application Serial No. 09/797,488

Appeal brief

Page 41

RELATED PROCEEDINGS APPENDIX

There are no other appeals or interferences related to the present application.